Custom Asymmetric Page Split Generalized Index Search Trees and Formal Concept Analysis

Ben Martin and Peter Eklund

School of Computer Science and Software Engineering
The University of Wollongong
Northfields Avenue, Wollongong, NSW 2522, Australia
monkeyiq@users.sourceforge.net
peklund@uow.edu.au

Abstract. This paper investigates the scalability of applying Formal Concept Analysis to large data sets. In particular we present enhancements based on an existing spatial data structure, the RD-Tree, to better support both specific use with Formal Concept Analysis as well as generic multidimensional applications. Our experiments are motivated by the application of Formal Concept Analysis to a virtual filesystem [11,20,16]. In particular the libferris [1] Semantic File System.

1 Introduction: Information Retrieval and Formal Concept Analysis

In previous work we have shown that the application of spatial indexing to Formal Concept Analysis (FCA) can vastly improve query times [19]. Subsequent research was directed toward improving the spatial indexing techniques themselves [18]. This paper improves upon [18] by applying FCA to improve the spatial indexing structure.

The primary goal of this paper is to improve the efficiency of FCA on large formal contexts. Two subgoals can be seen – the improvement of the spatial indexing structure independent of the data it is indexing and improvements that rely on both the spatial indexing structure and the fact that FCA is being applied using that spatial index. An example of the latter would be the spatial index relying on knowledge from the FCA application in order to employ specialized compression as explained in Section 4.

FCA [10] is a well understood technique of data analysis. FCA takes as input a binary relation I between two sets normally referred to as the object set O and attribute set A and produces a set of "Formal Concepts" which are a minimal representation of the natural clustering of the input relation I. Formal concepts are hereafter referred to simply as concepts. A concept is a pair $(X \subseteq O, Y \subseteq A)$ such that X cannot be enlarged without reducing $|Y|$ and vice versa. The application of FCA to non-binary relations, such as a table in a relational database,

S.O. Kuznetsov and S. Schmidt (Eds.): ICFCA 2007, LNAI 4390, pp. 82–97, 2007.

can be achieved by first transforming or "scaling" the input data into a binary relation [10,21].

For a concept (X, Y), X is called the extent and is the set of all objects that have all of the attributes in Y, similarly Y is called the intent and is the set of all attributes possessed in common by all the objects in X. As the number of attributes in Y increases, the concept becomes more specific, i.e. a specialization ordering is defined over the concepts of a formal context by:

$$(X_1, Y_1) \leq (X_2, Y_2) :\Leftrightarrow Y_2 \subseteq Y_1$$

This ordering is a concept lattice which is normally presented as a Hasse diagram with special labeling rules [10].

A common approach to document and information retrieval using FCA is to convert associations between many-valued attributes and objects into binary associations between the same objects O and new attributes A. For example, modeling a filesystem the files would form the object set O. A many-valued attribute showing a file's size as numeric data may be converted into three attributes: `small`, `medium`, `large` which are then associated with the same set of files O. The binary relation I between $o \in O$ and $a \in A = \{$small, medium, large$\}$ is formed by asserting oIa depending on the level of the numeric size value of file o. The binary relation I is referred to as a formal context in FCA.

This is the approach adopted in the ZIT-library application developed by Rock and Wille [23] as well as the Conceptual Email Manager [6]. The approach is mostly applied to static document collections (such as news classifieds) as in the program Rfca [5] but also to dynamic collections (such as email) as in Mail-Sleuth [2] and files in the Logical File System (LISFS) [20]. In all but the latter two the document collection and full-text keyword index are static. Thus, the FCA interface consists of a mechanism for dynamically deriving binary attributes from a static full-text index. Many-valued contexts are used to materialize formal contexts in which objects are document identifiers.

A specialized form of information retrieval system is a virtual file system [11,20,16]. The idea of using FCA to generate a virtual filesystem was first proposed by using a logical generalization of FCA [8,7] and in more recent work using an inverted file index and generating the lattice closure as required by merging inverted lists [20]. In a virtual file system scalability becomes a critical concern because such a system deals with potentially millions of documents and hundreds/thousands of attributes [19,17].

It has been found that spatial indexing structures can greatly reduce typical query times in FCA [19] – we will discuss the type of query we consider in the next section. This has prompted research into improving the existing spatial indexing structures to better support FCA. In many cases the improvements needed by FCA are also applicable to general purpose multidimensional queries. As such, our empirical testing includes application to generic data mining input as well as specific application to FCA. The core focus of the paper remains on improving RD-Trees with the express purpose of improved FCA performance.